C hapter 3

Design of Sequential Circuits

Design Steps

- Read the problem statement carefully. You may be given the state diagram and state table for the circuit which will be the easiest case.
- If not given to you, derive the state diagram and state table for the circuit from the problem statement.
- See if you can reduce them as we learnt in the previous chapter.
- Count the number of states in the state diagram (call it N) and calculate the number of flip-flops needed (call it P) by solving the equation $2^{P-1} < N \leq 2^{P}$. This is best solved by guessing the value of P.
- Assign a unique P-bit binary number to each state. Often, the first state = 0, the next state = 1, etc.

- Derive the state transition table and the output table.
- Decide on the types of flip-flops to use. In many cases it is easier to use all JK's.
- Derive the input table for each flip-flop using the excitation tables for the type (we will study them in next section).
- Derive the input equations for each flip-flop based as functions of the input and current state of all flip-flops.
- Summarize the equations by writing them in one place.
- Draw the circuit diagram. Most homework assignments will not go this far, as the circuit diagrams are hard to draw neatly.

Excitation tables

- To excite a flip-flop, means to operate it.
- This table is important in the analysis procedure to deduce the next state of the flip-flop according to the current value of output and the flip-flops inputs.
- In the design procedure, we want to make the reverse operation.
- We want the circuit to make a specific response, and we need to determine the possible inputs to flip-flops accordingly.

D flip-flop Excitation table

- **If the current state is 0 and we want it to be 0, we input D=0.**
- **If the current state is 0 and we want it to be 1, we input D=1 as the value inputted on D input will appear on the Q output in the next clock pulse whatever the value of Q(t).**
- **So, the column of D is the same as Q(t+1) directly regardless the value of Q(t).**

Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

- **We can write that D = Q(t+1).**

RS flip-flop Excitation table

Q(T)	Q(T+1)	S	R
0	0		
0	1		
1	0		
1	1		

S	R	Q(T+1)
0	0	Q(t) No change
0	1	0 Reset
1	0	1 Set
1	1	? not used

JK flip-flop Excitation table
T flip-flop Excitation table

(Report)

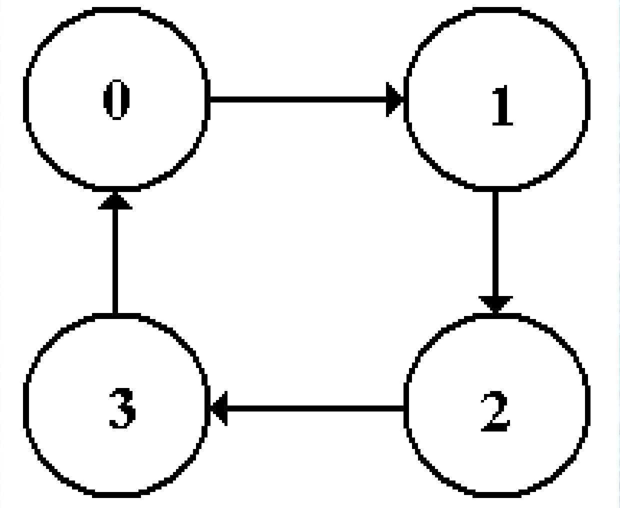
Try to deduce the values in the table as done in the SR above

Q(T)	Q(T+1)	J	K
0	0		
0	1		
1	0		
1	1		

Q(t)	Q(t+1)	T
0	0	
0	1	
1	0	
1	1	

Design Problem: A Modulo-4 Counter

- This counter counts from 0 to 3.
- When the direction is not specified, we usually intend to build a modulo-four up-counter: 0, 1, 2, 3, 0, 1, 2, 3, etc.
- **Step 1: Derive the state diagram and state table for the circuit.**



Present State	Next State
0	1
1	2
2	3
3	0

- **Step 2:** **Count the number of states in the state diagram (call it N) and calculate the number of flip-flops needed (call it P) by solving the equation $2^{P-1} < N \leq 2^P$.**
- modulo-4 counter has four states: labeled 0, 1, 2, and 3. We solve the equation $2^{P-1} < 4 \leq 2^P$ by noting that $2^1 = 2$ and $2^2 = 4$, so we have determined that $2^1 < 4 \leq 2^2$, hence P = 2.

- **Step 3:** **Assign a unique P-bit binary number to each state. Often, the first state = 0, the next state = 1, etc.**

State	2-bit state
0	0 0
1	0 1
2	1 0
3	1 1

- **Step 4:** **Derive the state transition table and the output table.**

Present State		Next State
0	00	01
1	01	10
2	10	11
3	11	00

- **Step 5:** **Decide on the types of flip-flops to use. When in doubt, use all JK’s.**

- **Step 6:** **Derive the input table for each flip-flop using the excitation tables for the type.**

Q(t)	Q(t+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

PS		NS		Flip-flop Input	
Y_1	Y_0	Y_1	Y_0	J_1	K_1
0		0		0	x
0		1		1	x
1		1		X	0
1		0		X	1

PS		NS		Input			
Y_1	Y_0	Y_1	Y_0	J_1	K_1	J_0	K_0
0	**0**	0	**1**	0	x	**1**	**x**
0	**1**	1	**0**	1	x		
1	**0**	1	**1**	x	0		
1	**1**	0	**0**	x	1		

Step 7: Derive the input equations for each flip-flop based as functions of the input and current state of all flip-flops.

Here is this author’s set of rules to match an expression to a given column.

1- If a column does not have a 0 in it (i.e. all 1's and x's), then it is equal to constant value 1.

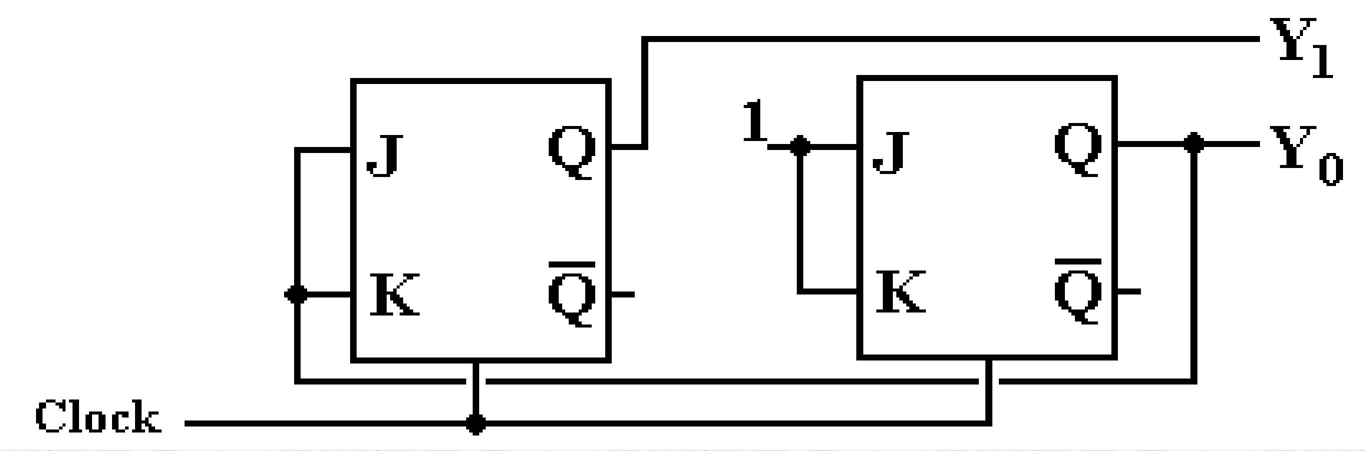
2- If a column does not have a 1 in it (i.e. all 0's and x's), match it is equal to constant value 0.

3- If the column has both 0’s and 1’s in it, try to match it to a single variable from the present state variables. Only the 0’s and 1’s in a column must match the suggested function.

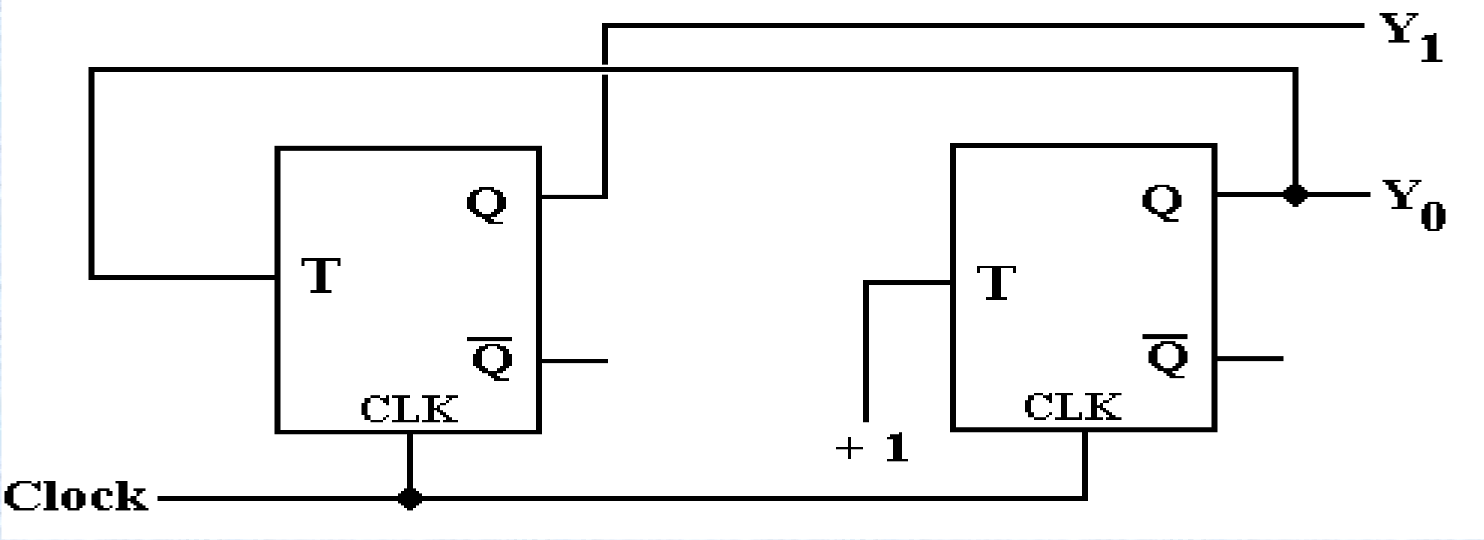
4- If every 0 and 1 in the column is a mismatch, match to the complement of a column.

5- If all the above fails, try for simple combinations of the present state.

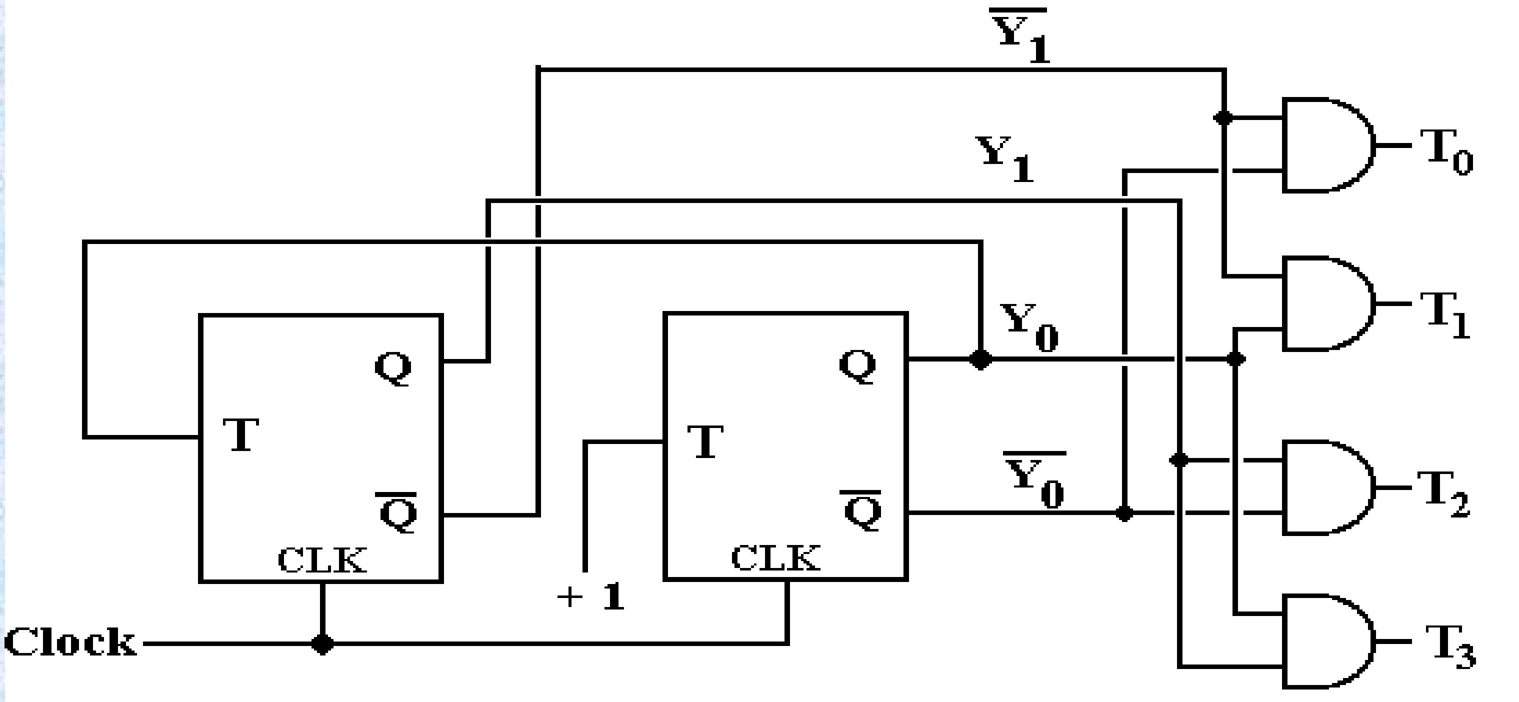
- **Step 8: Summarize the equations by writing them in one place.**
- Here they are.
- **$J_1 = Y_0$ $K_1 = Y_0$**
- **$J_0 = 1$ $K_0 = 1$**
- This is a counter, so there is no Z output.
-
- **Step 10: Draw the circuit diagram.**

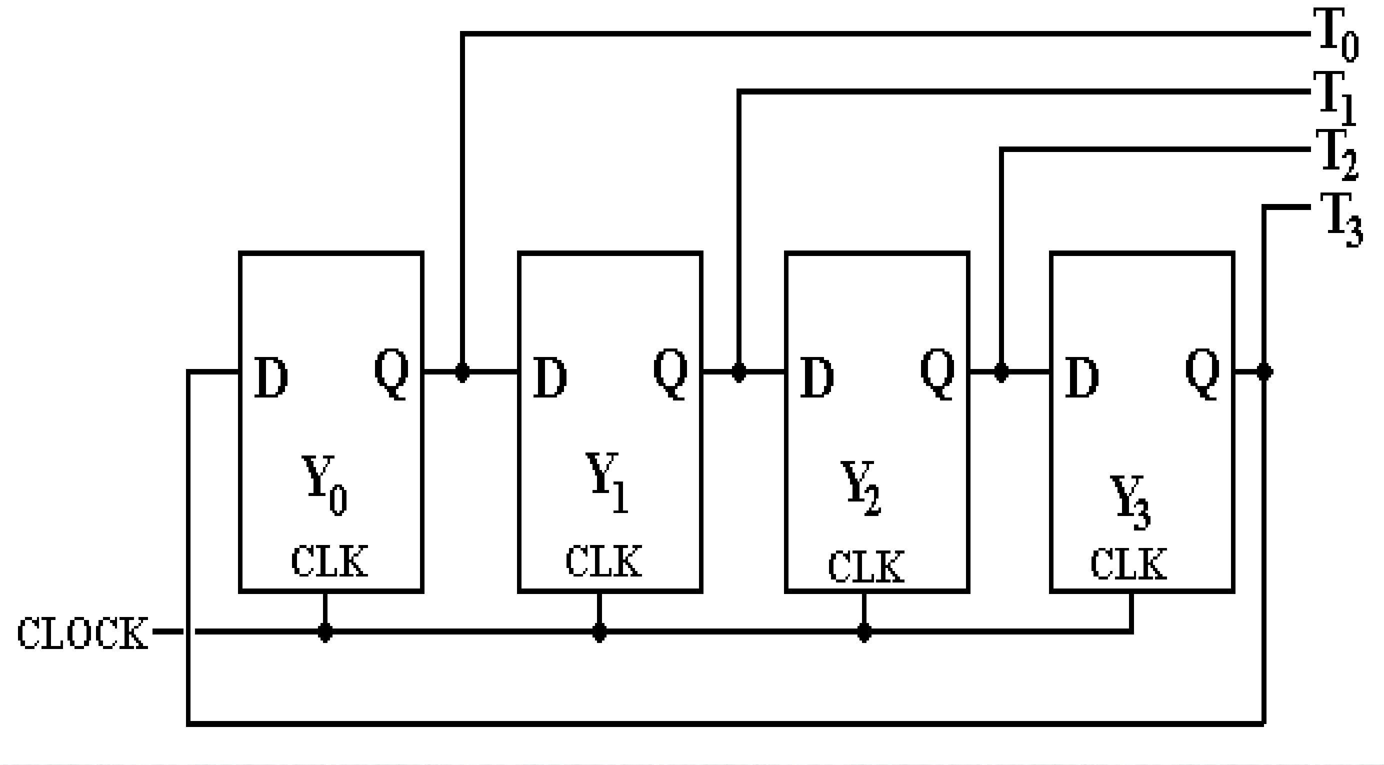


- . Recall that a JK flip-flop can be used to emulate a T flip-flop by setting the J input equal to the K input.
- Note that the design has the following interesting property.
- $J_1 = K_1 = Y_0$
- $J_0 = K_0 = 1$
- Given this, we can replace each JK flip-flop with a T flip-flo



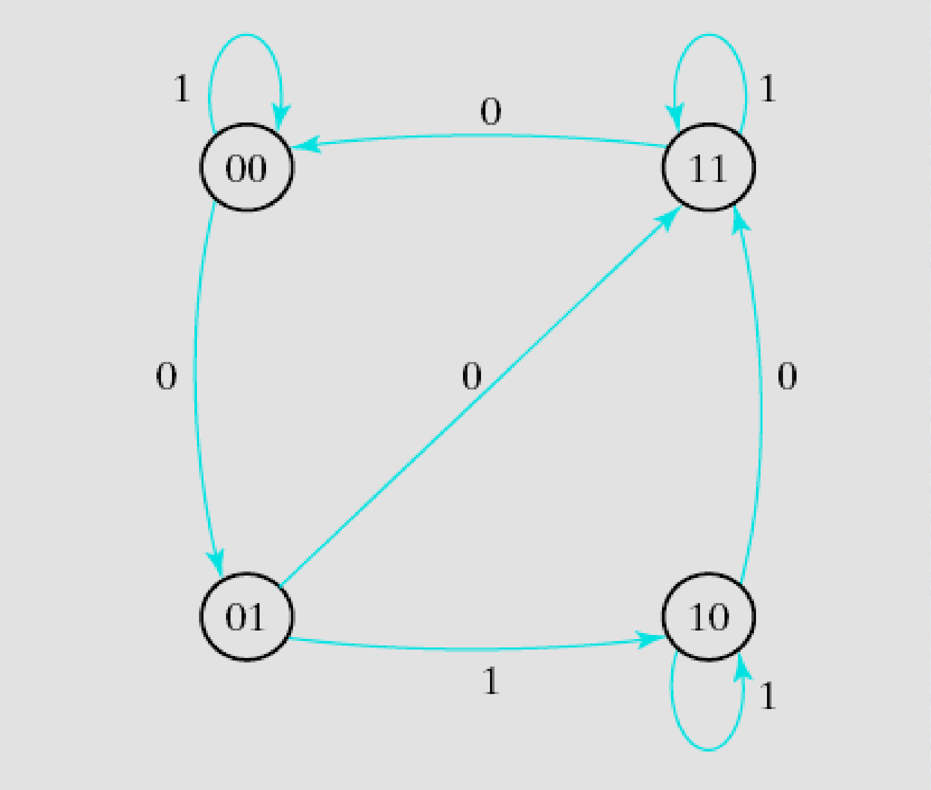
- . A more realistic circuit would output discrete pulses corresponding to the decoded output, so that first $T_0 = 1$ and all others are 0, then $T_1 = 1$ and all others are 0, etc.
- In order to produce the discrete signals T_0, T_1, T_2, and T_3, we need to add a decoding phase to the counter.




T0
T1
T2
T3
D
Q
Y0
CLK
D
Q
Y1
CLK
D
Q
Y2
CLK
D
Q
Y3
CLK
CLOCK

Example 2:

Given the following state diagram, make a design for such circuit.



Present State		Input	Next State	
A	B	X	A	B
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

Step 6: Decide on the types of flip-flops to use.

We will use D flip-flop as he did not mention a specific type.

Step 7: Derive the input table for each flip-flop using the excitation tables for the type.

Present State		Input	Next State		Flip-flop inputs	
A	**B**	**X**	**A**	**B**	D_A	D_B
0	0	0	0	1	0	1
0	0	1	0	0	0	0
0	1	0	1	1	1	1
0	1	1	1	0	1	0
1	0	0	1	1	1	1
1	0	1	1	0	1	0
1	1	0	0	0	0	0
1	1	1	1	1	1	1

- **Step 8:** **Derive the input equations for each flip-flop based as functions of the input and current state of all flip-flops.**
- From the table and in terms of A, B and X, DA and DB are:
- $\mathbf{D_A(A,B,X) = \Sigma(2,3,4,5,7)}$
- $\mathbf{D_B(A,B,X) = \Sigma(0,2,4,7)}$
-
- Simplifying using the map, we get:
- $\mathbf{D_A(A,B,X) = A'B + AB' + BX = A \oplus B + BX}$
- $\mathbf{D_B(A,B,X) = A'X' + B'X' + ABX}$

Report

Try to implement the same circuit with JK and T flip-flops and compare between the circuits in complexity.

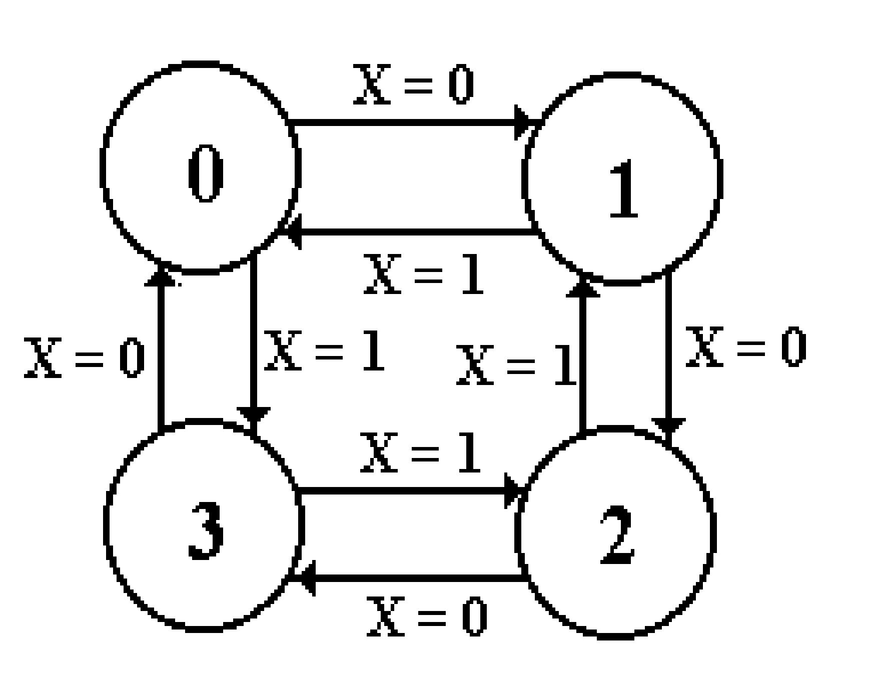
Example: The Modulo-4 Up-Down Counter

- For the next design, we introduce a problem that uses input. This is a modulo-4 up-down counter.
- The input X is used to control the direction of counting.
- If X = 0, the device counts up:

  0, 1, 2, 3, 0, 1, 2, 3, etc.

- If X = 1, the device counts down:

  0, 3, 2, 1, 0, 3, 2, 1, etc.



Step 1: **Derive the state diagram and state table for the**

Step 2: **Count the number of states in the state diagram and calculate the number of flip-flops needed.**

The number of states are 4 so we need 2 flip-flops.

Step 3: **Assign a unique binary number to each state.**

Present State	Next State	
	X = 0	**X = 1**
0	1	3
1	2	0
2	3	1
3	0	2

State	**2-bit Vector**
0	0 0
1	0 1
2	1 0
3	1 1

Step 4: Derive the state transition table and the output table.

Step 5: Build the state table for the problem.

Present State		**Input**	Next State	
A	**B**	**X**	**A**	**B**
0	0	0	0	1
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

Step 6: Decide on the types of flip-flops to use.

We choose to use two JK flip-flops as it produces less complicated circuit.

Step 7: Derive the input table for each flip-flop using the excitation tables for the type.

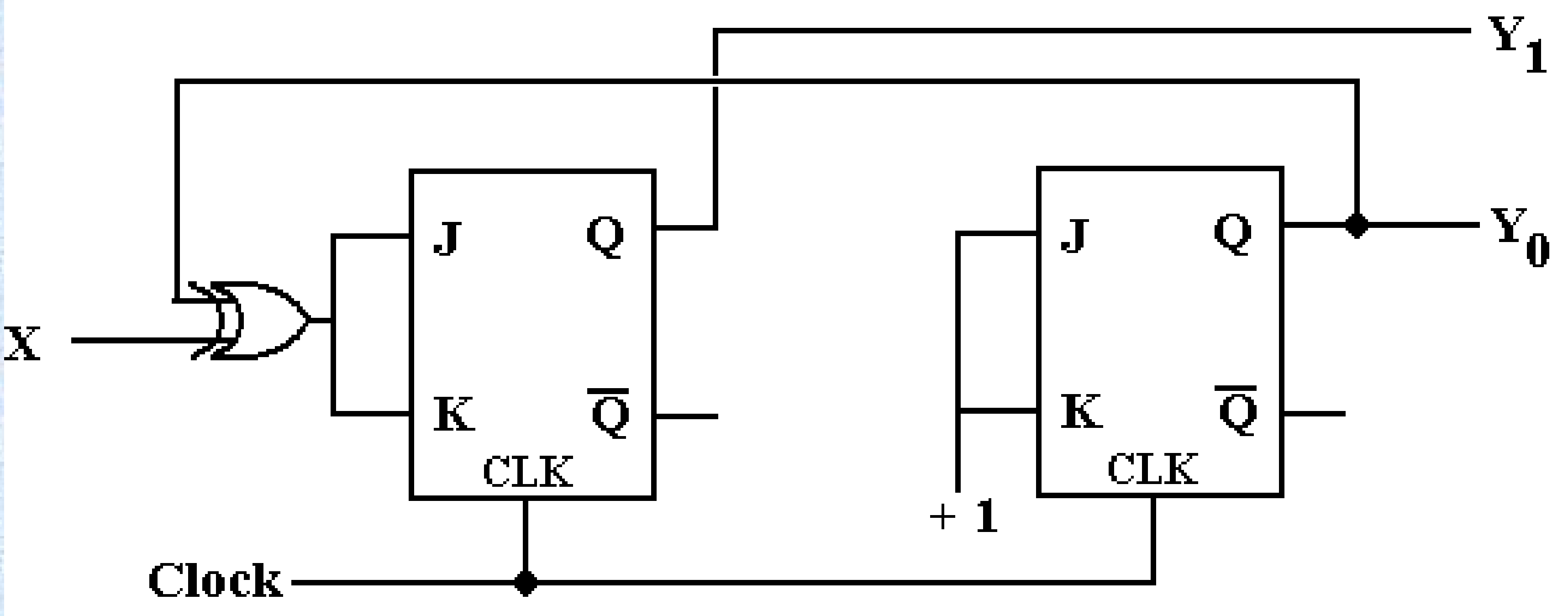
PS		Input	NS		Flip-flop inputs			
A	**B**	**X**	**A**	**B**	**JA**	**KA**	**JB**	**KB**
0	0	0	0	1	0	X		
0	0	1	1	1	1	X		
0	1	0	1	0	1	X		
0	1	1	0	0	0	X		
1	0	0	1	1	X	0		
1	0	1	0	1	X	1		
1	1	0	0	0	X	1		
1	1	1	1	0	X	0		

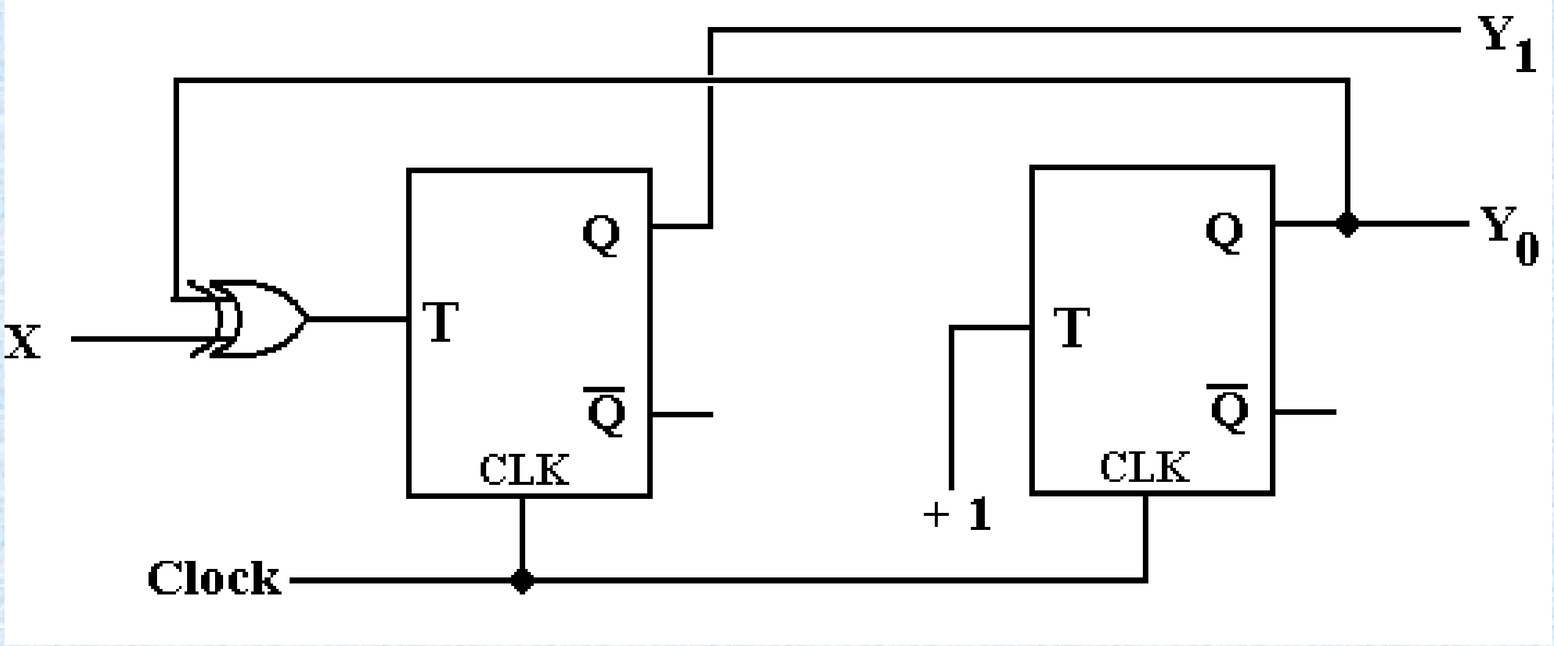
Step 8: Derive the input equations for each flip-flop based as functions of the input and current state of all flip-flops.

Use the 3-variable Karnaugh map to simplify the JA and KA to get:

$$J_A = X \oplus B \qquad K_A = X \oplus B$$

$$J_B = 1 \qquad K_B = 1$$



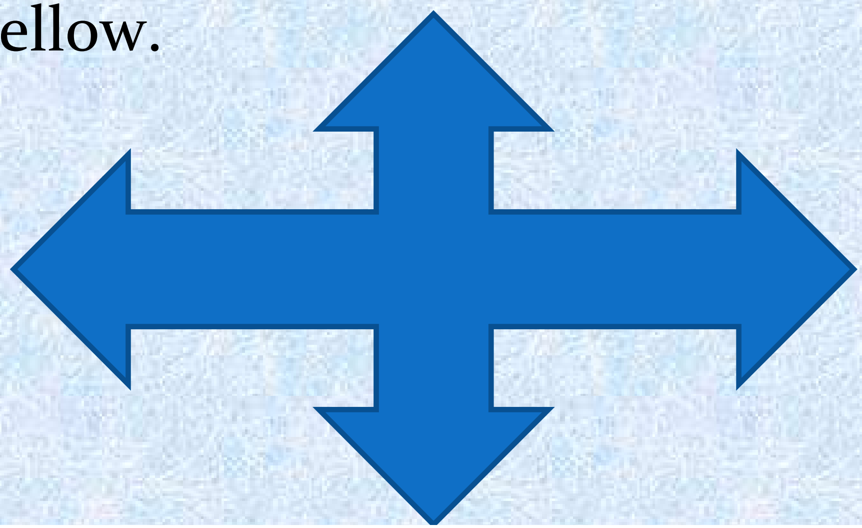


Y_1
Q
T
X
$\overline{Q}$
CLK
Clock
Q
Y_0
T
$\overline{Q}$
CLK
+ 1

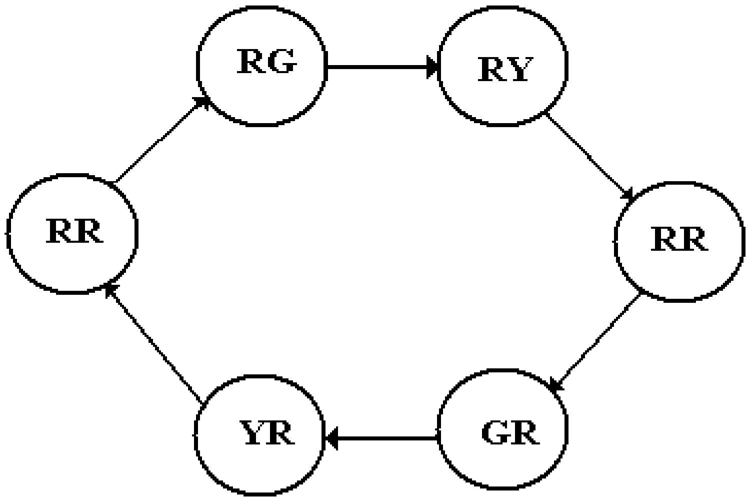
The Traffic Light Problem

- READ IT YOURSELF
- The light is at the intersection of two roads, one (NS) running North-South and one (EW) running East-West. The light should be considered as two coupled traffic lights, one called L1 and the other L2.
- Each of the lights (or pairs of lights) displays the standard sequence: Red, Green, Yellow.
- We see that there are

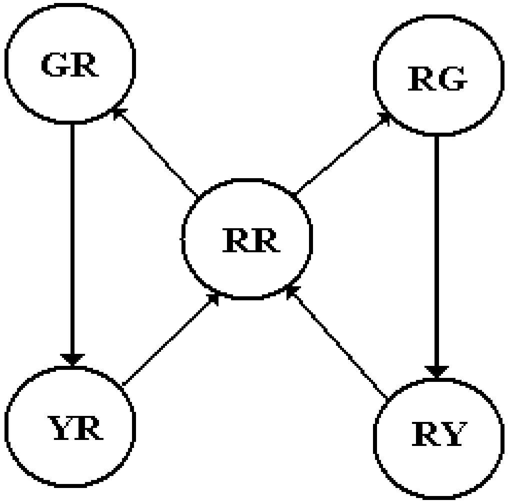
  six states in the system.



State	Light 1	Light 2	Symbol
0	Red	Red	RR
1	Red	Green	RG
2	Red	Yellow	RY
3	Red	Red	RR
4	Green	Red	GR
5	Yellow	Red	YR



A Six-State Design



A Five-State Design

PS		NS	
Number	Symbol	Number	Symbol
0	RR	1	RG
1	RG	2	RY
2	RY	3	RR
3	RR	4	GR
4	GR	5	YR
5	YR	0	RR

	Symbol	**$Q_2Q_1Q_0$**	**R1**	**G1**	**Y1**	**R2**	**G2**	**Y2**
0	RR	0 0 0	1	0	0	1	0	0
1	RG	0 0 1	1	0	0	0	1	0
2	RY	0 1 0	1	0	0	0	0	1
3	RR	0 1 1	1	0	0	1	0	0
4	GR	1 0 0	0	1	0	1	0	0
5	YR	1 0 1	0	0	1	1	0	0
6	RR	1 1 0	1	0	0	1	0	0
7	RR	1 1 1	1	0	0	1	0	0